TDE B - Asteroide

Vencimento 10 de dez de 2023 por 23:59	**Pontos** 60	**Enviando** um upload de arquivo	**Tipos de arquivo** zip, rar, e 7z	
Disponível em 10 de dez de 2023 em 23:59				

Esta tarefa foi travada 10 de dez de 2023 em 23:59.

Introdução

Esta tarefa tem por objetivo desenvolver um programa em assembly de 8086 do jogo **Asteroid Way**, conforme as especificações deste documento.

O programa deverá ser desenvolvido individualmente ou em duplas. Não serão aceitos equipes com mais de 2 integrantes. Não caso de duplas, um aluno ou nenhum em um dos 10 grupos criados na opção de menu Pessoas, ata TDE B.

O trabalho deverá ser apresentado no último dia de aula de correção (8 de dezembro) pela equipe. A apresentação deverá ser realizada utilizando o **GUI Turbo Assembler** (Windows) ou **Turbo Assembler** (Máquina Virtual). Portanto, o programa deve ser testado frequentemente em ambos dois montadores. Somente após a apresentação, a equipe poderá submeter o trabalho.

Asteroide

O jogo Asteroide tem por objetivo navegar no espaço evitando ou destruindo os asteroides que podem danificar a nave. O jogo tem uma nave no centro da tela. A partir do lado direito da tela, asteroides surgem em direção para o lado esquerdo da tela (devido a impressão que a nave está indo para frente). O jogador pode mover a nave para cima ou para baixo assim como pode ainda ou destruir o asteroide atirando raio (barra de espaço). A nave possui um nível de saúde que a cada colisão com um asteroide é reduzido (limite = 10 colisões). Caso o nível de saúde da nave zerar, o jogo termina. Em cada fase, em separador de saúde (entre pouco a nível cheio) é enviado aleatoriamente quando o nível de saúde estiver abaixo de 50%. A partir do nível 2, um escudo pode ser pego pela nave. Tal que a colisão de um asteroide não danifica a nave (nível de saúde). O escudo ocorre a cada 10 segundos. Cada etapa dura um determinado tempo. Ao passar para a próxima etapa, a velocidade dos asteroides aumenta.

Requisitos

Esta seção apresenta os requisitos do jogo.

Vídeo

O programa deve utilizar o modo de vídeo 320 x 200 (modo de vídeo = 13H). Neste modo de vídeo, a tela é dividida em 320 colunas e 200 linhas, conforme mostra a Figura 1. Cada pixel ocupa um byte na memória, podendo assumir, portanto, uma dentre 256 cores distintas (2⁸ possibilidades = 256 cores). A memória de vídeo nesse modo ocupa, então, 320 * 200 bytes, ou 64000 bytes.

Figura 1. Estrutura do modo de vídeo 320 x 200

Como o vídeo utiliza o método de mapeamento de E/S em memória, parte da memória principal é reservada para o vídeo. O segmento de memória de vídeo usado pelo modo 13h carrega na posição A000h. Para plotar um pixel na tela, basta escrever nesse deslocamento a cor desejada para o pixel, conforme a Tabela 1. Para evitar um pixel na posição 320, 0, por exemplo, na coluna 320, linha 0, basta escrevermos a cor desejada na posição 100 * 16* 320 do segmento de memória A000H (a endereço inicial da memória de vídeo é A000H:0000H).

Tabela 1 - Código das cores.

Hexadecimal	Cor	Hexadecimal	Cor
0	Preto	8	Cinza escuro
1	Azul	9	Azul claro
2	Verde	A	Verde claro
3	Ciano	B	Ciano claro
4	Vermelho	C	Vermelho claro
5	Magenta	D	Magenta claro
6	Marrom	E	Amarelo
7	Cinza claro	F	Branco

Apesar que a interrupção 10H (vídeo) realizar a escrita de pixels na tela, a operação de escrita através do endereçamento de memória (endereçamento indireto) é mais eficiente do que via interrupção. A cada disco que a interrupção deve ajustar a CPU para todos. Tipo, entre "A CPU está ocupada neste momento, mas no você deixar seu cotno após o final - não entremova em cartuda." Além disso, é possível utilizar instruções de manipulação de strings para ausilar na escrita. Por isso, deve-se utilizar endereçamento indireto via instruções de manipulação de strings para a realização de escrita de pixels (no caso de escrito de texto, interrupções podem ser utilizadas).

Tela Inicial

A tela inicial deve apresentar 3 elementos, conforme ilustrado na Figura 2:

- **Nome do Jogo**: deve se imprimir o nome "Asteroid Way" usando o gerador de arte em ASCII em http://patorjk.com/software/taag no cor verde em duas linhas.
- **Elementos do Jogo**: deve se mostrar a nave, o asteroide, escudo e tiro (PROJ) do programa (por exemplo, movimentação dos asteroides, controle do tiro, entre outras).
- **Menu**: deve disponibilizar as opções **Jogar** e **Sair**. Cada opção deve ser listada pelos caracteres 1 e 2, respectivamente. Com o caso deles ser para cima e para baixo, o usuário deverá controlar a seleção da opção, representada pela movimentação dos colchetes. A tecla **Enter** executa a respectiva opção.

Figura 2 - Exemplo da tela inicial do jogo

O Jogo

A tela do jogo é composta por 5 elementos:

- **Status do jogo**: deve mostrar o jogador informado do seu tempo de jogo (barra azul à esquerda) e o nível de saúde (barra verde à direita), como ilustrado na Figura 3. Deve estar localizado na parte inferior da tela. A cada segundo, a barra azul deve diminuir. Quando o jogador colidir com um asteroide, a barra verde deve diminuir 1/10.

Figura 3 - Status.

- **Nave**: o programa deve implementar uma nave com pelo menos 2 cores, conforme ilustrado na Figura 4. Cada nave deve ocupar 10 pixels de altura e 10 pixels de largura. A nave deve iniciar centralizada horizontalmente e verticalmente na tela. Ao testar de seta para cima e seta baixo devem ser utilizadas para movimentar a nave. Cada vez que a nave colidir com um asteroide, o status de saúde diminui 1/10 (quando chegar a zero, deve terminar o jogo). A nave pode atirar pressionando a barra de espaço.

Figura 4 - Exemplos da nave.

- **Asteroides**: os asteroides devem ter o formato ilustrado na Figura 6 (10 pixels de altura e 10 pixels de largura). Eles aparecem aleatoriamente na tela direita da tela e movem-se para o lado esquerdo. A rotina de geração de números aleatórios deve ser buscada na internet (a fonte deve ser referenciada no trabalho). Os asteroides devem desaparecer ao colidir com a nave ou ao chegar ao lado esquerdo da tela.

Figura 6 - Representação do asteroide e escudo.

- **Projétil**: são disparados pela nave ao pressionar a tecla barra de espaço. O projétil deve ser um pixel branco e deve mover-se para a direita no dobro da velocidade dos demais. O projétil destrói o asteroide ao acertar qualquer um dos seus pixels.

Tela do Fim do Jogo

No caso do nível de saúde zerar, a tela deve ser preenchida (atraso entre cada pixel) até o status do jogo com a cor vermelha e o texto em branco "V O C E P E R D E U !" utilizando o gerador de arte em ASCII deve ser mostrado. Caso o usuário passou de todas as fases, a tela do jogo deve ser preenchida com a cor amarela (atraso entre cada pixel) até o status do jogo e o texto em branco "P A R A B É N S !" utilizando o gerador de arte em ASCII deve ser mostrado. O status do jogo não deve ser pior.

Restrições

Além dos requisitos descritos, as seguintes restrições devem ser atendidas:

1. A impressão (ou ocultação) da nave, asteroides, saúde, escudo e tiro devem ser realizados exclusivamente via endereçamento indireto ou com instruções de manipulação de strings.
2. O código não deve possuir duplicatas de rotinas (por exemplo, uma rotina de movimentação para os asteroides e outro para as ações, sendo que é possível fazer somente uma).
3. O tempo entre as movimentações dos elementos do jogo (50 ms) e o tempo de cada etapa devem ser configuráveis.
4. O programa deve estar rodando quando montado utilizando o **GUI Turbo Assembler** (Windows) ou **Turbo Assembler** (Máquina Virtual).

Entregas

A submissão do trabalho deve ser realizada via este site de uma arquivo compactado (por ex: .zip) com 2 arquivos:

1. **Código fonte do programa**: deve ser somente o arquivo com extensão ASM. Caso for enviado o executável, será descontado 15 pontos da nota final.
2. **Relatório do trabalho**: deve ser submetido um arquivo no **formato PDF** (o trabalho escrito utilizando o formato da Revista Scientific Linux Industry). Não é necessário incluir o título em inglês e abstract. A formatação também fará parte da avaliação do trabalho. Caso for enviado o documento em Word ou LibreOffice, será descontado 0,5 ponto da nota final. Cada grupo deve produzir um relatório com pelo menos três seções:
   - **Jogo**: esta seção deve descrever em linhas gerais o funcionamento do jogo - com o apoio de figuras que tratem das 3 elementos: tela inicial, tela do jogo e tela final. Deve ser realizadas qualquer funcionalidade requisitada pelo trabalho que não tenha sido relatada plenamente o que foi requisitado. Não é para entrar em detalhes do código nesta seção. É como se fosse um manual de operação do jogo.
   - **Solução**: deve descrever como é a lógica do programa em detalhes e o algoritmo do programa. Esta seção pode ter o suporte de diversas figuras (devidamente identificadas e referenciadas no texto) incluindo em fluxograma da solução que facilite a compreensão tanto do algoritmo como do código.
     - Seção 2.1 **Algoritmo**: deve descrever o fluxo principal do programa. Esta seção deve ter um fluxograma do funcionamento do laço do jogo. Além disso, esta seção deve mostrar o **tamanho do código em bytes** e o **número de linhas de código**.
     - Seção 2.2 **Memória**: deve descrever as principais variáveis (ou blocos de memória) utilizadas no programa. Além disso, esta seção deve mostrar o **tamanho da memória utilizada** (seja dados e pilha código (explique o processo utilizada para estimar o tamanho de cada memória (dados e código).
     - Seção 2.3 **Rotinas**: deve listar e descrever as principais rotinas (**PROCs**) do programa (por exemplo, movimentação dos asteroides, controle do tiro, entre outras). No caso da rotina de geração de números aleatórios, especificar o origem do programa ou algoritmo que você utilizou (lembre-se que ele formato da revista, exista a seção de **Referências**).
     - Seção 2.4 **Instruções 8086**: esta seção somente existe se você utilizar alguma instrução da arquitetura 8086 que não foi vista em aula (ou nos exercícios). Explique a necessidade da instrução e como ela funciona.
   - **Conclusões**: o grupo deve discutir as dificuldades encontradas no processo e o que aprendeu com o trabalho.
   - **Referências**: caso você utilizou algum site ou livro para alguma operação, você deve listar aqui (conforme a ABNT) o(s)s site(s).

Dicas

Esta seção tem por objetivo passar algumas dicas que visam facilitar a implementação do trabalho.

Escrita de Strings com Cor

Para escrever uma string, pode-se utilizar a função 13H da interrupção 10H.

```
AH = 13h
Entrada:
    bit 6: 1- atualiza a posição do cursor após a escrita
    bit 1: 1- string contém caracteres e atributos
DL = atributo do caractere se o bit 1 em AL for 0
DH = número da página de vídeo
CX = Tamanho, em caracteres, da string
CX = Tamanho, em caracteres, da string
ES:BP = endereço do início da string
```

Note esta interrupção reconhece os códigos CR, LF, e backspace.

Suspensão da Execução por um Determinado Intervalo de Tempo

A interrupção 15H oferece diversos tipos de serviços. Dentre eles, existe a função (AH=86H) de suspender a execução de programa que a invoca por um determinado intervalo de tempo. O intervalo é definido em microsegundos (1/1.000.000 de segundo) pelos registradores CX e DX, onde CX possui os 16 bits mais significativos e DX possui os 16 bits menos significativos. Como 1 segundo é igual a 1.000.000 microsegundos, para um intervalo de 2 segundos, o registrador CX deve ter o valor 1EH e o registrador DX deve ser o valor 8480H (pois, 1E8480H = 2.000.000D).

Leitura de Teclado sem Aguardar

A interrupção 16H oferece diversas funções relacionadas ao teclado. No caso da leitura de uma tecla, é necessário que o programa não espere por uma tecla. Para isso, a interrupção 16H oferece um serviço de verificar o status do teclado (AH=01H), isto é, se existe algumas tecla no buffer do teclado. Caso não houver inserirtado no buffer, o Zero Flag (ZF) é setado. Para buscar a tecla do buffer, deve-se utilizar o serviço AH=00H da mesma interrupção. Além de trazer o código da tecla, ela remove a mesma do buffer do teclado.

Asteroid Way			
Critérios	Avaliações		Pts
Tela de Entrada - Formatação Formatação da tela de entrada conforme descrito	2 pts Pontos máximos	0 pts Nenhum ponto	2 pts
Tela de Entrada - Menu Funcionamento das opções do menu	2 pts Pontos máximos	0 pts Nenhum ponto	2 pts
Tela do Jogo - Inicial Apresentação inicial da tela do jogo apresenta os 5 elementos conforme descritos (características e localização)	2 pts Pontos máximos	0 pts Nenhum ponto	2 pts
Tela do Jogo - Movimentação da nave A nave é movimentada pelas teclas seta para cima e seta para baixo.	4 pts Pontos máximos	0 pts Nenhum ponto	4 pts
Tela do Jogo - Tiro da Nave O projétil disparado pela nave percorre um caminho horizontal até algum obstáculo ou limite esquerdo da tela. Ele é vivível a cada mudança de posição.	3 pts Pontos máximos	0 pts Nenhum ponto	3 pts
Tela do Jogo - Asteroides Asteroides são gerados em posições aleatórias e movimentam-se para a esquerda. Eles são vivíveis a cada mudança de posição.	4 pts Pontos máximos	0 pts Nenhum ponto	4 pts
Tela do Jogo - Detecção de colisão Existe a detecção de colisão levando com que o asteroide desapareça da tela.	3 pts Pontos máximos	0 pts Nenhum ponto	3 pts
Tela do Jogo - Tiro no Asteroide Asteroide desaparece ao ser atingido por tiro da nave.	2 pts Pontos máximos	0 pts Nenhum ponto	2 pts
Tela do Jogo - Reparador Um reparador de saúde é enviado aleatoriamente quando o nível de saúde estiver abaixo de 50%.	2 pts Pontos máximos	0 pts Nenhum ponto	2 pts
Tela do Jogo - Reparação Saúde retorna a 100% ao colidir com nave.	2 pts Pontos máximos	0 pts Nenhum ponto	2 pts
Tela do Jogo - Escudo Há implementação do escudo com a geração aleatória e movimentação.	2 pts Pontos máximos	0 pts Nenhum ponto	2 pts
Tela do Jogo - Proteção do escudo Ao colidir com o escudo, a nave deve fazer inverso da colisão por 5 segundos.	2 pts Pontos máximos	0 pts Nenhum ponto	2 pts
Tela do Jogo - Status do Jogo Barra de saúde e tempo funcionam corretamente	3 pts Pontos máximos	0 pts Nenhum ponto	3 pts
Tela do Jogo - Mudança de nível Aumento da velocidade ao passar de nível	2 pts Pontos máximos	0 pts Nenhum ponto	2 pts
Tela do Jogo - Fim do Jogo Jogo termina quando a saúde chega a zero.	2 pts Pontos máximos	0 pts Nenhum ponto	2 pts
Tela do Fim do Jogo Funcionalidade conforme descrito.	3 pts Pontos máximos	0 pts Nenhum ponto	3 pts
Técnico - Código Programa está bem documentado e mostra um bom estilo	1 pts Pontos máximos	0 pts Nenhum ponto	1 pts
Técnico - Operação Programa opera corretamente sob condições normais (sem a), não tem bugs	2 pts Pontos máximos	0 pts Nenhum ponto	2 pts
Técnico - Robtras O programa utiliza rotinas genéricas (por exemplo, desenho dos pacas, checagem de jogo, entre outras). Não possui repetição de blocos de instruções.	2 pts Pontos máximos	0 pts Nenhum ponto	2 pts
Técnico - Acesso à Memória O programa utiliza instruções de endereçamento indireto e de manipulação de strings para o acesso à memória.	4 pts Pontos máximos	0 pts Nenhum ponto	4 pts
Técnico - Uso de Memória Uso de memória de forma eficiente (baixa quantidade necessária de memória).	2 pts Pontos máximos	0 pts Nenhum ponto	2 pts
Técnico - Configurável Uso de memória e constantes para as configurações do jogo (tempo, velocidade, do jogo, ...)	1 pts Pontos máximos	0 pts Nenhum ponto	1 pts
Relatório - Formatação e Gramática Formatação correta e escrita do texto atende as regras da português.	2 pts Pontos máximos	0 pts Nenhum ponto	2 pts
Relatório - Conteúdo Texto segue o que foi requisitado	3 pts Pontos máximos	0 pts Nenhum ponto	3 pts
		Total de pontos: 60	